

# JavaScriptまとめ

---

## 要素の切り替え・付与・削除

ボタンの表示を変更する（切り替える）。

### html

```
<div class="container">
  <div id="switch-class">button</div>
</div>
```

### CSS

```
.container {
  display: flex;
  justify-content: center;
  align-items: center;

  width: 100%;
  height: 100vh;
}

#switch-class {
  padding: 1rem 2rem;
  font-size: 2rem;
  font-weight: 700;
  text-transform: uppercase;
  background-color: #eee;
  border-radius: .5rem;
  cursor: pointer;
}

/* ID要素にクラスを付与して効果を切り替える場合、CSSの優先度の都合でこういう書式になる。
*/
#switch-class.add-appearance {
  color: #fff;
  background-color: brown;
}
```

要素を切り替えるとは、 ・『クリックしたイベント』を契機にオブジェクトや状況の変化を演出する。 ・ターゲットとなる要素へ変化させる効果を設定したクラスを付与する。

[契機をつくる]

クリックして関数を作動 => `EventTarget.addEventListener("click", 関数)`

```
// ベタベタな書き方 その1
document.getElementById("switch-class").addEventListener("click",
function() {
  this.textContent = "hello";
})
// 無名関数で『this』は使えない。だからベタベタの書き方になる。
document.getElementById("switch-class").addEventListener("click", () => {
  document.getElementById("switch-class").textContent = "hello";
})
```

行数は増えるがインスタンスにして運用した方が見通しがよいと思われるので、当面はこの書式で進める。

```
const switchClass = document.getElementById("switch-class")
switchClass.addEventListener("click", () => {
  switchClass.textContent = "hello"
})
```

**クラスの付与 => TargetObject.classList.add("class\_name\_string"), TargetObject.className = "class\_name\_string"**

```
// ID要素にクラスを付与して効果を切り替える場合、CSSの優先度に留意する。
const switchClass = document.getElementById("switch-class")
switchClass.addEventListener("click", () => {
  switchClass.textContent = "hello"

  switchClass.classList.add("add-appearance")
  または、
  switchClass.className = "add-appearance"
})
```

**クラスの削除 => TargetObject.classList.remove("class\_name\_string")**

```
const switchClass = document.getElementById("switch-class")
switchClass.addEventListener("click", () => {
  switchClass.textContent = "hello"
  switchClass.classList.remove("add-appearance")
})
```